INTEGRATED INFORMATION SUPPORT SYSTEM (IISS)
Volume V - Common Data Model Subsystem
Part 27 - Distributed Request Supervisor Product Specification

J. Althoff, M. Apicella
Control Data Corporation
Integration Technology Services
2970 Presidential Drive
Fairborn, OH 45324-6209

September 1990

Final Report for Period 1 April 1987 - 31 December 1990

Approved for Public Release; Distribution is Unlimited
NOTICE

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever, regardless whether or not the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data. It should not, therefore, be construed or implied by any person, persons, or organization that the Government is licensing or conveying any rights or permission to manufacture, use, or market any patented invention that may in any way be related thereto.

This technical report has been reviewed and is approved for publication.

DAVID L. JUDSON, Project Manager
WRDC/MTI
Wright-Patterson AFB, OH 45433-6533

FOR THE COMMANDER:

BRUCE A. RASMUSSEN, Chief
WRDC/MTI
Wright-Patterson AFB, OH 45433-6533

If your address has changed, if you wish to be removed from our mailing list, or if the addressee is no longer employed by your organization please notify WRDC/MTI, Wright-Patterson Air Force Base, OH 45433-6533 to help us maintain a current mailing list.

Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.
This document establishes the design of Function DRS, Distributed Request Supervisor, one of the major functions of the Configuration Item, to be built and formally accepted by the ICAM program office.

**BLOCK 11:**

**INTEGRATED INFORMATION SUPPORT SYSTEM**

Vol V - Common Data Model Subsystem

Part 27 - Distributed Request Supervisor Product Specification
FOREWORD

This technical report covers work performed under Air Force Contract F33600-87-C-0464, DAPro Project. This contract is sponsored by the Manufacturing Technology Directorate, Air Force Systems Command, Wright-Patterson Air Force Base, Ohio. It was administered under the technical direction of Mr. Bruce A. Rasmussen, Branch Chief, Integration Technology Division, Manufacturing Technology Directorate, through Mr. David L. Judson, Project Manager. The Prime Contractor was Integration Technology Services, Software Programs Division, of the Control Data Corporation, Dayton, Ohio, under the direction of Mr. W. A. Osborne. The DAPro Project Manager for Control Data Corporation was Mr. Jimmy P. Maxwell.

The DAPro project was created to continue the development, test, and demonstration of the Integrated Information Support System (IISS). The IISS technology work comprises enhancements to IISS software and the establishment and operation of IISS test bed hardware and communications for developers and users.

The following list names the Control Data Corporation subcontractors and their contributing activities:

<table>
<thead>
<tr>
<th>SUBCONTRACTOR</th>
<th>ROLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control Data Corporation</td>
<td>Responsible for the overall Common Data Model design development and implementation, IISS integration and test, and technology transfer of IISS.</td>
</tr>
<tr>
<td>D. Appleton Company</td>
<td>Responsible for providing software information services for the Common Data Model and IDEF1X integration methodology.</td>
</tr>
<tr>
<td>ONTEK</td>
<td>Responsible for defining and testing a representative integrated system base in Artificial Intelligence techniques to establish fitness for use.</td>
</tr>
<tr>
<td>Simpact Corporation</td>
<td>Responsible for Communication development.</td>
</tr>
<tr>
<td>Structural Dynamics Research Corporation</td>
<td>Responsible for User Interfaces, Virtual Terminal Interface, and Network Transaction Manager design, development, implementation, and support.</td>
</tr>
<tr>
<td>Arizona State University</td>
<td>Responsible for test bed operations and support.</td>
</tr>
</tbody>
</table>

iii
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>SCOPE</td>
<td>1-1</td>
</tr>
<tr>
<td>1.1</td>
<td>Identification</td>
<td>1-1</td>
</tr>
<tr>
<td>1.2</td>
<td>Functional Summary</td>
<td>1-1</td>
</tr>
<tr>
<td>2.0</td>
<td>DOCUMENTS</td>
<td>2-1</td>
</tr>
<tr>
<td>2.1</td>
<td>Reference Documents</td>
<td>2-1</td>
</tr>
<tr>
<td>2.2</td>
<td>Terms and Abbreviations</td>
<td>2-1</td>
</tr>
<tr>
<td>3.0</td>
<td>REQUIREMENTS</td>
<td>3-1</td>
</tr>
<tr>
<td>3.1</td>
<td>Structural Description</td>
<td>3-1</td>
</tr>
<tr>
<td>3.2</td>
<td>Functional Flow</td>
<td>3-1</td>
</tr>
<tr>
<td>3.3</td>
<td>Interfaces</td>
<td>3-2</td>
</tr>
<tr>
<td>3.3.1</td>
<td>Inputs/Outputs</td>
<td>3-2</td>
</tr>
<tr>
<td>3.4</td>
<td>Program Interrupts</td>
<td>3-2</td>
</tr>
<tr>
<td>3.5</td>
<td>Timing and Sequencing Description</td>
<td>3-3</td>
</tr>
<tr>
<td>3.6</td>
<td>Special Control Features</td>
<td>3-3</td>
</tr>
<tr>
<td>3.7</td>
<td>Storage Allocation</td>
<td>3-3</td>
</tr>
<tr>
<td>3.7.1</td>
<td>Database Definition</td>
<td>3-3</td>
</tr>
<tr>
<td>3.7.1.1</td>
<td>File Description</td>
<td>3-3</td>
</tr>
<tr>
<td>3.7.1.2</td>
<td>Table Description</td>
<td>3-3</td>
</tr>
<tr>
<td>3.7.1.3</td>
<td>Item Description</td>
<td>3-3</td>
</tr>
<tr>
<td>3.8</td>
<td>Object Code Creation</td>
<td>3-3</td>
</tr>
<tr>
<td>3.9</td>
<td>Adaptation Data</td>
<td>3-3</td>
</tr>
<tr>
<td>3.10</td>
<td>Detail Design Description</td>
<td>3-4</td>
</tr>
<tr>
<td>3.10.1</td>
<td>Where Include File Used List</td>
<td>3-4</td>
</tr>
<tr>
<td>3.10.2</td>
<td>Where External Routine Used List</td>
<td>3-6</td>
</tr>
<tr>
<td>3.10.3</td>
<td>Main Program Parts List</td>
<td>3-7</td>
</tr>
<tr>
<td>3.10.4</td>
<td>Module Documentation</td>
<td>3-9</td>
</tr>
<tr>
<td>3.10.5</td>
<td>Include File Description</td>
<td>3-18</td>
</tr>
<tr>
<td>3.10.6</td>
<td>Hierarchy Chart</td>
<td>3-27</td>
</tr>
<tr>
<td>3.11</td>
<td>Program Listings Comments</td>
<td>3-29</td>
</tr>
<tr>
<td>4.0</td>
<td>QUALITY ASSURANCE PROVISIONS</td>
<td>4-1</td>
</tr>
<tr>
<td>4.1</td>
<td>Introduction and Definitions</td>
<td>4-1</td>
</tr>
<tr>
<td>4.2</td>
<td>Computer Programming and Test</td>
<td>4-1</td>
</tr>
</tbody>
</table>
SECTION 1
SCOPE

1.1 Identification

This specification establishes the design of Function DRS, "Distributed Request Supervisor", one of the major functions of the Configuration Item, to be built and formally accepted by the ICAM Program Office. This CI constitutes one of the subsystems of the Common Data Model Processor (CDMP).

1.2 Functional Summary

The overall objectives of this CPCI are to:

1. Determine the appropriate sequence of inter database Join, Union and Outer Join operations required to produce the result for a multi-database transaction.

2. Coordinate and control the interactions among a user's application process (AP), the generated Request Processor (RP) and the Aggregator(s) for both single and multi-database transactions.
SECTION 2
DOCUMENTS

2.1 Reference Documents


2.2 Terms and Abbreviations

Attribute Use Class: (AUC)

Conceptual Schema: (CS)

Common Data Model Processor: (CDMP)

Common Data Model: (CDM) Describes common data application process formats, form definitions, etc., of the IISS and includes conceptual schema, external, internal schemas, and schema transformation operators.

Data Field: (DF) An element of data in the external schema. It is by this name that an NDML programmer reference data.

Database Management System: (DBMS)

Distributed Request Supervisor: (DRS) This IISS CDM subsystem configuration item controls the execution of distributed NDML queries and non distributed updates.

Domain: A logical definition of legal attribute class values.
Domain Constraint: Predicate that applies to a single domain.

**External Schema:** (ES)

Forms: Structured views which may be imposed on windows or other forms. A form is composed of fields where each field is a form, item, or window.

**Forms Processor:** (FP) A set of callable execution time routines available to an application program for form processing.

**Internal Schema:** (IS)

Integrated Information Support System: (IISS) A test computing environment used to investigate, demonstrate, and test the concepts of information management and information integration in the context of Aerospace Manufacturing. The IISS addresses the problems of integration of data resident on heterogeneous databases supported by heterogeneous computers interconnected via a local Area Network.

**Mapping:** The correspondence of independent objects in two schemas: ES to Cs or CS to IS.

**Network Transaction Manager:** (NTM) Performs the coordination, communication, and housekeeping functions required to integrate the application processes and system services resident on the various hosts into a cohesive system.

**Neutral Data Manipulation Language:** (NDML) A language developed by the IISS project to provide uniform access to common data, regardless of database manager or distribution criteria. It provides distributed retrieved and single node updates.

**ORACLE:** Relational DBMS based on the SQL (Structured Query Language, a product of ORACLE Corp., Menlo Park, CA). The CDM is an ORACLE database.

**Parcel:** A sequential file containing sections source code of the input application program.

**Request Processor:** (RP) A COBOL program that will satisfy a retrieval or update NDML subtransaction against a particular Database Management System.

**User Interface:** (UI) Controls the user's terminal and interfaces with the rest of the system.

**Virtual Terminal Interface:** (VTI) Performs the interfacing between different terminals and the UI. This is done by defining a specific set of terminal features and protocols which must be supported by UI software which constitutes the Virtual Terminal Definition. Specific terminals are then mapped against the Virtual Terminal software by specific software modules written for each type of real terminal supported.
SECTION 3
REQUIREMENTS

3.1 Structural Description

A graphic portrayal of this CPCI is included in Section 3.10. This chart shows the hierarchical relationship of each module making up this CPCI.

The DRS has been coded as a COBOL subprogram with supporting subprograms.

It is internally composed of three subfunctions and defined in the DS Reference 8. These subfunctions are:

1. Initiate/Resume Subtransaction Processing
2. Schedule Stages
3. Initiate CS/ES Transform Processing

3.2 Functional Flow

This CPCI implements the logic defined in the Development Specification for this CPCI. Details of inputs/outputs and relationships between modules are to be found in Section 3.10.

This CPCI has been designated to operate in an interactive mode. It must operate in the system environment established for IISS; that is, use of the Network Transaction Manager.

The following exceptions to the Development Specification are noted:

1. The DS calls for the CS/ES transform step to be controlled by the DRS as a separate process, using file input-output and NTM interprocess communication. In interests of efficiency, the CS/ES transform is controlled by code generated into the user AP directly. This saves one file of external query results and allows the interaction with the CS/ES transform to be direct, without use of NTM services.

2. The DS called for the DRS to be a separate process. In the interests of efficiency, it has been implemented as a subprogram called from the user's AP.

3. The contents of Transmission Cost Table are compiled into the DRS. The DRS specifies that this be found in a file.
3.3 Interfaces

The following diagram depicts the interface of DRS and the other CPCI's.

```
+------+
| USER |
+------+
   |
   +------+
   | AP    |
+------+
   |
   +------+
   | SUBROUTINE CALLS |
+----------+
| DRS       |
| CS ES TRANSFORMERS |
+----------+
   |
   +------+
   | NTM SERVICES |
+---------+
 | REQUEST PROCESSORS |
| AGGREGATORS PS620141320 |
| FILE UTILITIES PS620141330 |
+---------+
```

3.3.1 Input/Outputs

The following table depicts the inputs and outputs of this CPCI. A detailed description for each item can be found in the DS for this CPCI.

<table>
<thead>
<tr>
<th>FUNCTION: DRS</th>
</tr>
</thead>
<tbody>
<tr>
<td>INPUT</td>
</tr>
<tr>
<td>---------------</td>
</tr>
<tr>
<td>OUTPUT</td>
</tr>
<tr>
<td>---------------</td>
</tr>
<tr>
<td>Subtransaction Number</td>
</tr>
<tr>
<td>DRS Action</td>
</tr>
<tr>
<td>Pool of Input Tables from the Users Application Process</td>
</tr>
<tr>
<td>CS Active List</td>
</tr>
<tr>
<td>Join Query Graph</td>
</tr>
<tr>
<td>Attribute Pair List</td>
</tr>
<tr>
<td>Results Field Table</td>
</tr>
</tbody>
</table>

3.4 Program Interrupts

The DRS makes use of NTM services to start and control multiple request processors at the same time. It also controls multiple instances of aggregators at the same time. It must wait until each process has completed before it can begin its next sequence of activities.
3.5 **Timing and Sequencing Description**

The DRS can control many request processors at the same time, asynchronously. In other words, it will start all subtransactions of a query and wait for them all to complete. When complete, it handles aggregation of these results. The aggregation may also execute asynchronously in parallel. The DRS will wait for all processes to complete. It has no time limits.

3.6 **Special Control Features**

Not applicable to this CPCI.

3.7 **Storage Allocation**

3.7.1 **Database Definition**

No databases are used by this CPCI.

3.7.1.1 **File Description**

No permanent files have been defined for this CPCI. It may use temporary scratch files for such things as generated program source code or temporary query results. The cost information table has not been implemented as a file.

3.7.1.2 **Table Description**

All tables used by this CPCI have been defined by the Development Specification for this CPCI.

3.7.1.3 **Item Description**

Not applicable to this CPCI.

3.8 **Object Code Creation**

The object code for this CPCI will be created by the system integration test team by using defined IISS Software Configuration Management procedures. This CPCI will use the COBOL language compiler.

3.9 **Adaptation Data**

This CPCI has been coded using ANSI COBOL. The intent was to provide a transportable system. Any system environment supporting this language, a virtual memory management scheme,
the COMM and NTM subsystems of IISS and the ORACLE Database Management System should be able to support this CPCI. Every possible attempt has been made to localize and identify any machine or environment dependent modules through the original design of the IISS and application of Configuration Management Procedures.

3.10 Detail Design Description

The following sections have been computer generated for this CPCI.

3.10.1 Where Include File Used List

The following lists each include file in the documentation group and all the modules documented in this specification which include them. The purpose of each module is listed as well.

DOCGROUP PS41310 Where-include-file-used List

<table>
<thead>
<tr>
<th>Include File</th>
<th>Module Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERRCDM</td>
<td>CDFUNC</td>
</tr>
<tr>
<td></td>
<td>CDLSC</td>
</tr>
<tr>
<td></td>
<td>CDS01</td>
</tr>
<tr>
<td></td>
<td>TOTOPN</td>
</tr>
<tr>
<td></td>
<td>TRMDML</td>
</tr>
<tr>
<td>CHKCDM</td>
<td>CDFUNC</td>
</tr>
<tr>
<td></td>
<td>TOTOPN</td>
</tr>
<tr>
<td></td>
<td>TRMDML</td>
</tr>
<tr>
<td>ERRPRO</td>
<td>CDFUNC</td>
</tr>
<tr>
<td></td>
<td>CDLSC</td>
</tr>
<tr>
<td></td>
<td>CDS01</td>
</tr>
<tr>
<td></td>
<td>TOTOPN</td>
</tr>
<tr>
<td></td>
<td>TRMDML</td>
</tr>
<tr>
<td>TCTABLE</td>
<td>CDS01</td>
</tr>
<tr>
<td>CITABLE</td>
<td>CDS01</td>
</tr>
<tr>
<td>RITABLE</td>
<td>CDS01</td>
</tr>
<tr>
<td>QITABLE</td>
<td>CDS01</td>
</tr>
<tr>
<td>SUBPROC</td>
<td>CDS01</td>
</tr>
<tr>
<td>RFTABLE</td>
<td>CDS01</td>
</tr>
<tr>
<td>STDRESP</td>
<td>CDS01</td>
</tr>
<tr>
<td>SRVRET</td>
<td>CDS01</td>
</tr>
</tbody>
</table>
DOCGROUP PS41310 Where-include-file-used List

<table>
<thead>
<tr>
<th>Include File</th>
<th>Module Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>FSMSG</td>
<td>CDS01</td>
</tr>
<tr>
<td>AGGMSG</td>
<td>CDS01</td>
</tr>
<tr>
<td>CSAL</td>
<td>CDS01</td>
</tr>
<tr>
<td>JQGTBL</td>
<td>CDS01</td>
</tr>
<tr>
<td>APL</td>
<td>CDS01</td>
</tr>
<tr>
<td>LNKEDG</td>
<td>CDS01</td>
</tr>
<tr>
<td>DUMPCIT</td>
<td>CDS01</td>
</tr>
<tr>
<td>DUMPRIT</td>
<td>CDS01</td>
</tr>
<tr>
<td>DUMPAPL</td>
<td>CDS01</td>
</tr>
<tr>
<td>DUMPRFT</td>
<td>CDS01</td>
</tr>
<tr>
<td>DUMPJQG</td>
<td>CDS01</td>
</tr>
<tr>
<td>DMPCSAL</td>
<td>CDS01</td>
</tr>
<tr>
<td>STDTyp</td>
<td>CDGTUSR</td>
</tr>
<tr>
<td>FPD</td>
<td>CDGTUSR</td>
</tr>
<tr>
<td>FPCODE</td>
<td>CDGTUSR</td>
</tr>
</tbody>
</table>

3-5
DOCGROUP PS41310 Where-include-file-used List

Include File  Module Name
---------  --------
NDDL       CDGTUSR
OK         CDGTUSR

3.10.2 Where External Routine Used List

The following lists each external function or routine in the documentation group and all the documented modules which call it. The purpose of each module is listed as well.

DOCGROUP PS41310 Where-external-routine-used List

System Module Name
------- ------
ERRPRO    CDFUNC
          CDLSC
          CDS01
          TOTOPN
          TRMDML
RPMAIN    CDLSC
SIGERR    CDS01
WTHST     CDS01
GETUSR    CDS01
CHKMSG    CDS01
RCV       CDS01
ASCTIM    CDS01
NSEND     CDS01
ISEND     CDS01
CDJS1     CDS01
CDUS1     CDS01
CDOJS1    CDS01
DELFIL    CDS01
DOCGROUP PS41310 Where-external-routine-used List

<table>
<thead>
<tr>
<th>System</th>
<th>Module Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIGABT</td>
<td>CDS01</td>
</tr>
<tr>
<td>OPENX</td>
<td>CDS01</td>
</tr>
<tr>
<td>TRMNAT</td>
<td>TOTOPN</td>
</tr>
<tr>
<td>STRNCMP</td>
<td>TRMDML</td>
</tr>
<tr>
<td>STRLEN</td>
<td>STRFILL</td>
</tr>
<tr>
<td>MEMCPY</td>
<td>CDGTUSR</td>
</tr>
</tbody>
</table>

3.10.3 Main Program Parts List

The following lists each Main Program in the documentation group and all the modules which are called either by that module itself or by any of the documented modules which it calls. It is possible for a non-main module to be listed more than once if it is called by multiple modules. The called modules, in this case known as program parts, are marked as to whether they are documented here. If so, the phrase "well-defined module" appears by the module name, if not it is an "external "routine". The Purpose of the Main Program module is listed as well.
<table>
<thead>
<tr>
<th>Main Pgm</th>
<th>Module</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDFUNC</td>
<td>ERRPRO</td>
<td>External routine</td>
</tr>
<tr>
<td>CDGTUSR</td>
<td>STRNCPY</td>
<td>External routine</td>
</tr>
<tr>
<td></td>
<td>STRLEN</td>
<td>External routine</td>
</tr>
<tr>
<td></td>
<td>MEMCPY</td>
<td>External routine</td>
</tr>
<tr>
<td></td>
<td>STRNCMP</td>
<td>External routine</td>
</tr>
<tr>
<td>CDLSC</td>
<td>ERRPRO</td>
<td>External routine</td>
</tr>
<tr>
<td></td>
<td>RPMAIN</td>
<td>External routine</td>
</tr>
<tr>
<td>CDS01</td>
<td>ERRPRO</td>
<td>External routine</td>
</tr>
<tr>
<td></td>
<td>SIGERR</td>
<td>External routine</td>
</tr>
<tr>
<td></td>
<td>CDGTUSR</td>
<td>External routine</td>
</tr>
<tr>
<td></td>
<td>WHTHST</td>
<td>External routine</td>
</tr>
<tr>
<td></td>
<td>GETUSR</td>
<td>External routine</td>
</tr>
<tr>
<td></td>
<td>CHKMSG</td>
<td>External routine</td>
</tr>
<tr>
<td></td>
<td>RCV</td>
<td>External routine</td>
</tr>
<tr>
<td></td>
<td>ASCTIM</td>
<td>External routine</td>
</tr>
<tr>
<td></td>
<td>CDLSC</td>
<td>External routine</td>
</tr>
<tr>
<td></td>
<td>NSEND</td>
<td>External routine</td>
</tr>
<tr>
<td></td>
<td>ISEND</td>
<td>External routine</td>
</tr>
<tr>
<td></td>
<td>CDFUNC</td>
<td>External routine</td>
</tr>
<tr>
<td></td>
<td>CDJS1</td>
<td>External routine</td>
</tr>
<tr>
<td></td>
<td>CDUS1</td>
<td>External routine</td>
</tr>
<tr>
<td></td>
<td>CDOJS1</td>
<td>External routine</td>
</tr>
<tr>
<td></td>
<td>DELFIL</td>
<td>External routine</td>
</tr>
<tr>
<td></td>
<td>SIGABT</td>
<td>External routine</td>
</tr>
<tr>
<td>INTFTN</td>
<td>STRFILL</td>
<td>External routine</td>
</tr>
<tr>
<td></td>
<td>STRNCPY</td>
<td>External routine</td>
</tr>
<tr>
<td>Main Pgm Name</td>
<td>Module Name</td>
<td>Module Type</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------</td>
<td>-------------</td>
</tr>
<tr>
<td>TOTOPN</td>
<td>ERRPRO</td>
<td>External routine</td>
</tr>
<tr>
<td></td>
<td>OPENX</td>
<td>External routine</td>
</tr>
<tr>
<td>TRMDML</td>
<td>ERRPRO</td>
<td>External routine</td>
</tr>
<tr>
<td></td>
<td>CDS01</td>
<td>External routine</td>
</tr>
<tr>
<td></td>
<td>TRMNAT</td>
<td>External routine</td>
</tr>
</tbody>
</table>

### 3.10.4 Module Documentation

The following documentation describes information which is specific to each individual module in the documentation group being documented in this specification. It provides a compact way of getting information that would be otherwise buried within each module's source code.

The specific items in this module documentation have the following meanings:

- **NAME:** Name of program Module.
- **PURPOSE:** Purpose of Module as detailed in the source code.
- **LANGUAGE:** Programming language source code is written in.
  - The choices are:
    - VAX-11 FORTRAN
    - C (I/S-1 Workbench 'C')
    - VAX-11 COBOL
- **MODULE TYPE:** Whether a Program, Subroutine, or Function.
- **SOURCE FILE:** Name of Source File from file specification.
- **SOURCE FILE TYPE:** Source File Extension from file specification.
- **HOST:** Whether this is a host-dependent routine (VAX or IBM) or blank if host-independent.
- **SUBSYSTEM:** IISS sub-system this file resides in.
| **SUBDIRECTORY:** | Sub-directory of that subsystem in which this file resides. |
| **DOCUMENTATION GROUP:** | Name of documentation group of which this source file is a member. |
| **DESCRIPTION:** | A description of the module as obtained from the source code. |
| **ARGUMENTS:** | The arguments with which this routine is called if it is a Subroutine or a Function. |
| **INCLUDE FILES:** | A list of all the files that are included into this module as well as their purposes. |
| **ROUTINES CALLED:** | Subroutines or Functions, either documented or external, called by this module, if any. |
| **CALLED DIRECTLY BY:** | The documented routines which call this module, if any. |
| **USED IN MAIN PROGRAM(S):** | The documented Main Programs which contain this module in their parts list according to the list in section 3.10.3. |

The Module Documentation is arranged alphabetically according to Module Name.

**DOCGROUP PS41310 Module Documentation**

**NAME:** CDFUNC  
**PURPOSE:** DETERMINE AP NAME GIVEN THE FUNCTION AND HOST  
**LANGUAGE:** VAX-11 COBOL  
**SOURCE FILE:** CDFUNC  
**SOURCE FILE TYPE:** COB  
**HOST:**  
**SUBSYSTEM:** CDM  
**SUBDIRECTORY:** CDMR

**DESCRIPTION:**  
---  
- PERFORM A TABLE LOOK UP BASED ON THE GIVEN HOST NAME AND THE FUNCTION DESIRED.  
RETURN THE PROPER AP NAME.  
---
ARGUMENTS:

---------
FUNCT-IN          DSPLY[X(10)]
HOST-IN           DSPLY[XXX]
TARGET-AP         DSPLY[X(10)]
RET-STATUS        DSPLY[X(5)]

INCLUDE FILES:

-----------
ERRCDM
CHKCDM
ERRPRO

ROUTINES CALLED:

--------
ERRPRO

NAME: CDLSC
PURPOSE: Dynamically call the rp-main and return to drs
LANGUAGE: VAX-11 COBOL
SOURCE FILE: CDLSC
SOURCE FILE TYPE: COB
HOST: VAX
SUBSYSTEM: CDM
SUBDIRECTORY: CDMR

DESCRIPTION:

---------
This routine is called from the drs (cds01) to do the cobol dynamic call to the rp-main. There is an ibmversion of this routine that is simply a stub, since the ibm cannot handle standard cobol that does dynamic calls.

10/2/89 - CHANGED TO ALWAYS CALL RPMAIN.C NO MATTER WHAT THE LANGUAGE. THIS IS SO WE WILL NOT LONGER HAVE ANY DYNAMIC CALLS.
ARGUMENTS:

---------
QP-MOD-NAME     DSPLY[X(5)]
LOG-CHAN        DSPLY[999]
RP-MSG-LTH      DSPLY[9(5)]
RP-MSG          RECRD
STD-RESPONSE    RECRD
RPMAIN-LANG     DSPLY[X]
RET-STATUS      DSPLY[X(5)]

INCLUDE FILES:

---------
ERRCDM
ERRPRO

ROUTINES CALLED:

---------
RPMAIN
ERRPRO

DOC GROUP PS41310 Module Documentation

NAME: CDS01
PURPOSE: THE DISTRIBUTED REQUEST SUPERVISOR
LANGUAGE: VAX-11 COBOL
SOURCE FILE: CDS01
SOURCE FILE TYPE: COB
HOST: VAX
SUBSYSTEM: CDM
SUBDIRECTORY: CDMR

DESCRIPTION:

---------
- THE DRS IS THE RUN TIME MONITOR
OF ALL RUN TIME PROGRAMS NECESSARY
TO FULFILL A NDML REQUEST.

MOD REL 2.3:
ADD SUPPORT FOR "IN-LINE CODE" WHICH MEANS ONE OF
THE REQUEST PROCESSORS MAY BE LINKED IN LOCALLY
AND ACCESSED BY A COBOL "DYNAMIC" CALL. ALSO
EACH AGGREGATOR WILL HAVE AN ADDITIONAL CALL-LEVEL
INTERFACE. ADD SUPPORT FOR USE OF THE OUTER-JOIN
AGGREGATOR INSTEAD OF THE NOT IN SET AGGREGATOR.
MOD 4/30/88 - R. E. STEWART - ADDED CODE TO HANDLE SQLFORMS
ARGUMENTS:

SS-NO-SUBTRANS     DSPLY[999]
DRS-ACTION          DSPLY[X]
SS-POOL             RECRD
CS-ACTION-LIST      RECRD
JQG                 RECRD
JQG-ATTRIBUTE-PAIR-LIST RECRD
USER-RFT            RECRD
CS-RESULTS-FILE     DSPLY[X(80)]
CS-RESULTS-COUNT    DSPLY[9(6)]
RET-STATUS          DSPLY[X(5)]

INCLUDE FILES:

TCTABLE
CITABLE
RITABLE
QITABLE
SUBPROC
RFTABLE
STDRESP
ERRCDM
Srvret
FSMSG
AGMSG
CSAL
JQGTBL
APL
ERRPRO
LNKEDGE
DUMPCIT
DUMPRIT
DUMPAPL
DUMPRFT
DUMPJQG
DMPCSAI

ROUTINES CALLED:

ERRPRO
SIGERR
CDGTUSR
WTHST
GETUSR
CHKMSG
RCV
ASCTIM
CDLSC
NSEND
ISEND
CDFUNC
CDJS1
CDUS1
CDQJS1
DELFIL
SIGABT

3-13
NAME: TOTOPN
PURPOSE: CONTROL OPENING OF TOTAL DB FILES
LANGUAGE: VAX-11 COBOL
SOURCE FILE: TOTOPN
SOURCE FILE TYPE: COB
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY: NDDL

DESCRIPTION:
------------
BY USING A GLOBAL REALM CONTAINING ALL FILES
CURRENTLY OPENED BY TOTAL OF THIS PROCESS AND
A LOCAL REALM OF FILES A PARTICULAR RP NEEDS,
ONLY THE NEW FILES NEED BE OPENED AND RECORDED
IN THE GLOBAL REALM TABLE. A SINGLE "OPENX"
CALL MAY BE ISSUED, AND EACH FILE SUCCESSFULLY
OPENED STORED IN THE GLOBAL REALM. IF ANY FILE
IS FOUND IN ERROR, A MESSAGE IS LOGGED.

ARGUMENTS:
----------
LOCAL-REALM RECRD
GLOBAL-REALM RECRD
TOTAL-STATUS DSPLY[X(4)]

INCLUDE FILES:
--------------
ERRCDM
CHKCDM
ERRPRO

ROUTINES CALLED:
-----------------
OPENX
ERRPRO
NAME: TRMDML
PURPOSE: TERMINATE USE OF NDML AND NTM
LANGUAGE: VAX-11 COBOL
SOURCE FILE: TRMDML
SOURCE FILE TYPE: COB
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY: CDMR

DESCRIPTION:
-------------
- THIS MODULE WILL BE USED TO SIGNAL END OF ANY
NDML COMMAND PROCESSING. IT WILL SEND A SPECIAL
CALL TO THE DRS, SO THAT IT CAN NOTIFY EACH
ACTIVE RP TO DO A CLOSE AND TERMINATE ITS
PROCESSING. WHEN THE DRS RETURNS AFTER EACH RP IS
DONE, NTM SERVICE TRMNAT WILL BE CALLED TO STOP THE
RUN. NOTE, THE USER WILL NOT NEED TO USE TRMNAT.

ARGUMENTS:
-------------
TERMINATION-STATUS DSPLY[X]

INCLUDE FILES:
-------------
CHKCDM
ERRCDM
ERRPRO

ROUTINES CALLED:
-------------
CDS01
TRMNAT
ERRPRO

NAME: INTFTN
PURPOSE: CONVERT INTEGER VALUE TO CHARACTER STRING **
LANGUAGE: VAX-11 FORTRAN
SOURCE FILE: INTFTN
SOURCE FILE TYPE: FOR
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY: CDMR

DESCRIPTION:
-------------
ARGUMENTS:

-----

NUMBER  I*4
CHAROT  CHAR

DESCRIPTION:

THIS IS A VERSION OF THE UI ROUTINE GTUINF, MODIFIED TO
USE THE GLOBAL USER NAME AND ROLE NAME STORED BY
THE BATCH VERSION OF NDDL. IT IS CALLED BY THE DRS
INSTEAD OF THE NTM GETUSR SERVICE WHICH DOESN'T WORK
WHEN THERE ARE MULTIPLE USERS ON THE NTM AT THE SAME
TIME. IF THE DRS IS USED BY SOME OTHER ROUTINE THAN
NDDL, THIS MAY NOT WORK SINCE THE GLOBAL USER AND ROL
NAME WONT BE SET UP, IF IT DOESNT BLOW-UP, THAT'S OK
SINCE THE DRS IS USING THIS AS A WORKAROUND TO A
CDM PATCH TO ALLOW MULTIPLE CDM'S ON THE SAME IISS INSTANCE.
CHECK FOR SPECIAL SQLFORMS FLAG AND SET RCODE TO 77777 IF
FOUND

SYNOPSIS

FORTRAN VOID CDGTUSR(USRNAM, USROLE, RCODE)
   CHAR USRNAM[];
   CHAR USROLE[];
   CHAR RCODE[];

DESCRIPTION

THIS MODULE WILL RETURN USER NAME AND USER ROLE TO CALLER

INPUTS:

NONE

OUTPUTS:

USRNAM - USER'S NAME
USROLE - USER'S ROLE
RCODE - RETURN CODE
ARGUMENTS:

-------------
USRNAM CHAR []
USROLE CHAR []
RCODE CHAR []

INCLUDE FILES:

------------
STDTYP
FPD
FPCODE
NDDL
OK

ROUTINES CALLED:

-------------
STRNCOPY
STRLLEN
MEMCPY
STRNCMP

DOCGROUP PS41310 Module Documentation

NAME: STRFILL
PURPOSE:
LANGUAGE: C
SOURCE FILE: STRFILL
SOURCE FILE TYPE: C
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY: CDMR

ARGUMENTS:

-------------
S CHAR []
T CHAR *
N INT

ROUTINES CALLED:

-------------
STRNCOPY
3.10.5 Include File Descriptions

The following list contains a purpose and description of each include file in the documentation group as specified in the source code. The language it is written in is also given.

DOCGROUP PS41310 Include File Description

FILE NAME: AGGMSG
PURPOSE: AGGREGATOR INPUT MESSAGE
LANGUAGE: VAX-11 COBOL

DESCRIPTION:
----------
CONTAINS THE FORMAT OF THE INPUT MESSAGE FOR THE CDMP AGGREGATORS

DESCRIPTION :-

AGGREGATOR INPUT MESSAGE FORMAT

NIS = NOT IN SET

DOCGROUP PS41310 Include File Description

FILE NAME: APL
PURPOSE: JOIN QUERY ATTRIBUTE PAIR LIST
LANGUAGE: VAX-11 COBOL

DESCRIPTION:
----------
CONTAINS INFORMATION ABOUT THE JOIN ATTRIBUTES FOR NDML SUBTRANSACTIONS

DOCGROUP PS41310 Include File Description

FILE NAME: CHKCDM
PURPOSE: IISS CDMP CHECK STATUS CODES
LANGUAGE: VAX-11 COBOL

DESCRIPTION:
----------
CONTAINS ALL STATUS CODES FOR THE CDMP MODULES

3-18
FILE NAME: CITABLE
PURPOSE: COST INFORMATION TABLE
LANGUAGE: VAX-11 COBOL

DESCRIPTION:
-------------------
THIS TABLE IS USED BY THE DRS TO TRACK COSTS OF POSSIBLE SUBTRANSACTIONS.

FILE NAME: CS
PURPOSE: DISPLAY CONTENTS OF THE COST INFORMATION TABLE
LANGUAGE: VAX-11 COBOL

DESCRIPTION:
-------------------

FILE NAME: CSAL
PURPOSE: CONCEPTUAL SCHEMA ACTION LIST
LANGUAGE: VAX-11 COBOL

DESCRIPTION:
-------------------
TABLE TO HOLD CONCEPTUAL DATA ABOUT THE REQUEST

NOTE!!!!!! This table is cloned in both cdpre5 and cdpre4 so any changes made to this structure needs to be made in these cloned versions. Clone version is CSALX for CDPRE4.

NOTE AGAIN Any changes to the CS-ACTION-ENTRY must be reflected in CDP10B in the C code generation section. The length of CS-STRING2 has been hard coded in the generated C code in paragraph 210-GEN-MOVE-OF-TABLES.

***** THE CONCEPTUAL SCHEMA ACTION LIST
DOCGROUP PS41310 Include File Description

FILE NAME: DMPCSAL
PURPOSE: DISPLAYS THE CONTENTS OF THE CS ACTION LIST
LANGUAGE: VAX-11 COBOL

DESCRIPTION:
-------------

DOCGROUP PS41310 Include File Description

FILE NAME: DUMPAPL
PURPOSE: DISPLAYS THE CONTENTS OF THE ATTRIBUTE PAIR LIST
LANGUAGE: VAX-11 COBOL

DESCRIPTION:
-------------

DOCGROUP PS41310 Include File Description

FILE NAME: DUMPJQG
PURPOSE: DISPLAY THE CONTENTS OF THE JQG TABLE
LANGUAGE: VAX-11 COBOL

DESCRIPTION:
-------------

DOCGROUP PS41310 Include File Description

FILE NAME: DUMPRFT
PURPOSE: DISPLAY THE CONTENTS OF THE RFT TABLE
LANGUAGE: VAX-11 COBOL

DESCRIPTION:
-------------
DOCgroup PS41310 Include File Description

FILE NAME: DUMPRIT
PURPOSE: DISPLAY THE CONTENTS OF THE RIT TABLE
LANGUAGE: VAX-11 COBOL

DESCRIPTION:

 -----------

DOCgroup PS41310 Include File Description

FILE NAME: ERRCDM
PURPOSE: ISSERROR STATUS CODES FOR CDMP MODULES
LANGUAGE: VAX-11 COBOL

DESCRIPTION:

 -----------
 CONTAINS ALL ERROR CODES USED BY CDMP MODULES FOR ERROR HANDLING


DOCgroup PS41310 Include File Description

FILE NAME: ERRPRO
PURPOSE: PROCESS ERROR INCLUDE FILE
LANGUAGE: VAX-11 COBOL

DESCRIPTION:

 -----------


DOCgroup PS41310 Include File Description

FILE NAME: FPCODE
PURPOSE: FORM PROCESSOR RETURN CODES
LANGUAGE: C

DESCRIPTION:

 -----------


DOCGROUP PS41310 Include File Description

FILE NAME: FPD
PURPOSE: FORM PROCESSOR DATA
LANGUAGE: C

DESCRIPTION:
-------------

DESCRIPTION
DATA DEFINITIONS FOR ALL FORM PROCESSOR (INCLUDING MONITOR) DATA.

DOCGROUP PS41310 Include File Description

FILE NAME: JQGTBL
PURPOSE: JOIN QUERY GRAPH TELLS HOW TO CONNECT SUBTRANSACTIONS
LANGUAGE: VAX-11 COBOL

DESCRIPTION:
-------------

DOCGROUP PS41310 Include File Description

FILE NAME: LNKEDGE
PURPOSE: DETERMINE DUPLICATE EDGES IN THE JQG
LANGUAGE: VAX-11 COBOL

DESCRIPTION:
-------------

DURING JQG COLLAPSING, DUPLICATE JQG ENTRIES MAY RESULT WITH DIFFERENT APL'S. THIS WILL BE EXECUTED AT THE END OF SENDS FOR A STAGE AND WILL FIND THE DUPLICATE EDGES AND HOOK THE APL'S TOGETHER BEFORE THE CIT IS REBUILT AT THE BEGINNING OF THE NEXT STAGE.

DOCGROUP PS41310 Include File Description

FILE NAME: NDDL
PURPOSE:
LANGUAGE: C

DESCRIPTION:
-------------
DOC GROUP PS41310 Include File Description

FILE NAME: OK
PURPOSE: GOOD RETURN CODE VALUE FOR UI
LANGUAGE: C

DESCRIPTION:
---------
DESCRIPTION

CONTAINS THE VALUE FOR A GOOD RETURN CODE FROM THE USER INTERFACE

DOC GROUP PS41310 Include File Description

FILE NAME: QITABLE
PURPOSE: REQUEST PROCESSOR INFORMATION TABLE
LANGUAGE: VAX-11 COBOL

DESCRIPTION:
---------
THIS TABLE WILL TRACK ALL ACTIVE REQUEST PROCESSORS FOR THE DRS.

QITABLE.INC

DOC GROUP PS41310 Include File Description

FILE NAME: RFTABLE
PURPOSE: THE RESULT FIELD TABLE
LANGUAGE: VAX-11 COBOL

DESCRIPTION:
---------
CONTAINS CONCEPTUAL SCHEMA INFORMATION ABOUT THE RESULTS OF AN NDML REQUEST

THE RESULT FIELD TABLE

WHEN CHANGING THE STRUCTURE OF THIS TABLE BE SURE TO CHANGE THE LAYOUT IN THE LINKAGE SECTION OF THE DRS (CDS01) WHICH WAS COPIED FROM THIS.
DOCGROUP PS41310 Include File Description

FILE NAME: RITABLE
PURPOSE: RIT- RELATION INFORMATION TABLE
LANGUAGE: VAX-11 COBOL

DESCRIPTION:

-----------

USED BY THE DRS TO KNOW ABOUT EACH RELATION
IN A TRANSACTION

THIS TABLE MUST HAVE THE SAME NUMBER OF OCCURS
AS THE SUBPROC.INC   SINCE THEY ARE PARALLEL
TABLES.

DOCGROUP PS41310 Include File Description

FILE NAME: SRVRET
PURPOSE: MESSAGE FOR THE FILE SEND UTILITY
LANGUAGE: VAX-11 COBOL

DESCRIPTION:

-----------

MESSAGE FORMAT FOR THE FILE SEND INPUT

DOCGROUP PS41310 Include File Description

FILE NAME: STDRESP
PURPOSE: WS DEFINITION FOR STANDARD STATUS VARIABLE
LANGUAGE: VAX-11 COBOL

DESCRIPTION:

-----------

THE STANDARD 'PROCESS COMPLETE' MESSAGE
DOCGROUP PS41310 Include File Description

FILE NAME: STDTYP
PURPOSE: STANDARD TYPE DEFINITIONS
LANGUAGE: C

DESCRIPTION:

DESCRIPTION
THIS FILE ENSURES THAT THE FOLLOWING STANDARD TYPES ARE AVAILABLE:

- FLOAT - SINGLE PRECISION FLOAT
- DOUBLE - DOUBLE PRECISION FLOAT
- LONG - 32 BIT (OR LARGER) SIGNED INTEGER
- LBITS - 32 BITS (OR MORE) FOR BIT MANIPULATION
- INT - NATURAL SIZE SIGNED INTEGER
- UNSIGNED - NATURAL SIZE UNSIGNED INTEGER
- BOOL - NATURAL SIZE LOGICAL (ZERO / NON-ZERO ONLY)
- SHORT - 16 BIT (OR LARGER) SIGNED INTEGER
- USHORT - 16 BIT (OR LARGER) UNSIGNED INTEGER
- BITS - 16 BITS (OR MORE) FOR BIT MANIPULATION
- CHAR - SINGLE MACHINE CHARACTER (REAL CHARACTERS ALWAYS POSITIVE)
- TINY - 8 BIT (OR LARGER) SIGNED INTEGER
- UTINY - 8 BIT (OR LARGER) UNSIGNED INTEGER
- TBITS - 8 BITS (OR MORE) FOR BIT MANIPULATION
- TBOOL - 8 BIT (OR LARGER) LOGICAL (ZERO / NON-ZERO ONLY)
- METACHAR - 16 BIT (OR LARGER) AUGMENTED CHARACTER (SIGNED)
- VOID - FUNCTION THAT RETURNS NO VALUE
- FORTRAN - STORAGE CLASS FOR FOREIGN (NON-C) ROUTINES OR C ROUTINES WHICH ARE CALLABLE FROM FOREIGN ROUTINES

SINCE NOT ALL COMPILERS SUPPORT USHORT, TINY, AND UTINY, THE FUNCTIONS USHORT(), TINY(), AND UTINY() SHOULD BE USED WHENEVER REFERENCING THEM.
IN ADDITION, THE FOLLOWING UTILITY MACROS ARE DEFINED:

- `LURSHIFT(N, B)` - UNSIGNED LONG RIGHT SHIFT
- `MAX(A, B)` - MAXIMUM OF A AND B
- `MIN(A, B)` - MINIMUM OF A AND B
- `ABS(A)` - ABSOLUTE VALUE OF A
- `STRASN(A, B)` - TRANSPORTABLE A = B FOR STRUCTURES
- `NULL` - NULL POINTER VALUE (0)
- `TRUE` - 1
- `FALSE` - 0
- `SUCCESS` - EXIT(SUCCESS) INDICATES SUCCESSFUL COMPLETION
- `FAILURE` - EXIT(FAILURE) INDICATES ERRORS

THE FOLLOWING SYMBOLS SHOULD BE DEFINED BASED ON THE COMPILER BEING USED:

- `USHORT` - COMPILER SUPPORTS UNSIGNED SHORT
- `TINY` - COMPILER TREATS CHAR AS SIGNED
- `UTINY` - CHAR IS SIGNED AND COMPILER SUPPORTS UNSIGNED CHAR
- `VOID` - COMPILER SUPPORTS VOID
- `FORTRAN` - COMPILER SUPPORTS FORTRAN
- `STRASN` - DEFINE APPROPRIATE MACRO
- `SUCCESS` - DEFINE APPROPRIATE VALUE IF NOT 0
- `FAILURE` - DEFINE APPROPRIATE VALUE IF NOT 1

DOCGROUP PS41310 Include File Description

FILE NAME: SUBPROC
PURPOSE: SUBTRANSACTION PROCESSES ID TABLE
LANGUAGE: VAX-11 COBOL
DESCRIPTION:

---

THIS TABLE MUST HAVE THE SAME NUMBER OF OCCURS AS THE RITABLE.INC AND QITABLE.INC SINCE THEY ARE PARALLEL TABLES.
FILE NAME: TCTABLE
PURPOSE: TRANSMISSION COST TABLE
LANGUAGE: VAX-11 COBOL

DESCRIPTION:
-------------
Holds relative cost of transmission/processing file transfers/joins on the network and is used as a basis of stager/scheduler optimization algorithms.

These are the experimental values for the TCT:

3.10.6 Hierarchy Chart

```
      +----------+----------+----------+----------+----------+
     | CDFUNC  | CDLSC    | TOTOPN   | TRMDML   | STRFILL  |
     +----------+----------+----------+----------+----------+
     | ERRPRO   | OPENX    | ERRPRO   | STRNCPY  |
     +----------+----------+----------+----------+

2

      +----------+
     | CDLSC    |
     +----------+
     | RMAIN    | ERRPRO   |
     +----------+----------+
```
CDFUNC.......1
CDGTUSR ......3
CDLSC.........2
CDS01 .........3
CHKMSG
ERRPRO
GETUSR
MEMCPY
OPENX
RPMAIN
SIGERR
STRFILL.......1
STRLN
STRNCMP
STRNCPY
TOTOPN.......1
TRMDML ......1
TRMNAT ........3
TRMDML
WHTHST
3.11 Program Listings Comments

This information is contained in the Module Descriptions in section 3.10.
SECTION 4
QUALITY ASSURANCE PROVISIONS

4.1 Introduction and Definitions

"Testing" is a systematic process that may be preplanned and explicitly stated. Test techniques and procedures may be defined in advance, and a sequence of test steps may be specified. "Debugging" is the process of isolation and correction of the cause of an error.

"Antibugging" is defined as the philosophy of writing programs in such a way as to make bugs less likely to occur and when they do occur, to make them more noticeable to the programmer and the user. In other words, as much error checking as is practical and possible in each routine should be performed.

4.2 Computer Programming Test and Evaluation

The quality assurance provisions for test consists of the normal testing techniques that are accomplished during the construction process. They consist of design and code walk-throughs, unit testing, and integration testing. These tests are performed by the design team. Structured design, design walk-through and the incorporation of "antibugging" facilitate this testing by exposing and addressing problem areas before they become coded "bugs."